UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/536,668	08/27/2005	Kazuaki Nakajima	202064-9001	9812

1131 7590 01/03/2008

MICHAEL BEST & FRIEDRICH LLP
Two Prudential Plaza
180 North Stetson Avenue, Suite 2000
CHICAGO, IL 60601

EXAMINER	
KHANNA, MADHU	
ART UNIT	PAPER NUMBER
4117	
MAIL DATE	DELIVERY MODE
01/03/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/536,668	NAKAJIMA, KAZUAKI	
	Examiner	Art Unit	
	MADHU KHANNA	4117	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *27 May 2005*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-21* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-21* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *05/27/2007* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☒ All b)☐ Some * c)☐ None of:

1.☒ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date *See Continuation Sheet*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

Continuation of Attachment(s) 3). Information Disclosure Statement(s) (PTO/SB/08), Paper No(s)/Mail Date :05/27/2005, 02/08/2007, 04/12/2007
.

DETAILED ACTION

1. This communication is in response to Application No. 10/536,668 filed on 27 May 05. The preliminary amendment filed May 27, 2005 which provides change to claims 5-6, 10, 15 and 20-21 is hereby acknowledged.

Information Disclosure Statement

2. Regarding IDS filed 2/08/07, the references cited in the European Search Report EP 03 77 4230, with the exception of US 2002/155848 AI (SURYANARAYANA LALITHA), and US 5,956,027 (Krishnamurthy Balachander) have been considered, but will not be listed on any patent resulting from this application because they were not provided on a separate list in compliance with 37 CFR 1.9.8(a)(1). The above mentioned reference was cited with either an invalid publication number or inventor. In order to have the references printed on such resulting patent, a separate listing, preferably on a PTO/SB/08A and 08B form, must be filed within the set period for reply to this Office action.

Claim Rejections - 35 USC § 101

3. Claim 1 is are rejected under 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

4. Claim 1 is rejected under 35 U.S.C. 101 as being directed to non-statutory subject matter.

In this case, claim 1, seems to be directed to a "system" comprising several machines, i.e. a "server", a "prescribed terminal" and "other terminal", as such, it is not directed to an eligible patentable category subject matter recited in 35 U.S.C. 101 (i.e., process, machine, manufacture, or composition of matter) but rather a combination of machines. For the purposes of examination, the claim will be treated as an apparatus/machine claim.

5. Claims 13-15 are rejected under 35 U.S.C. 101 as being directed to non-statutory subject matter.

In this case, claims 13-15 seem directed to a program, as such it is noted that computer-related inventions whether descriptive or functionally descriptive material are non-statutory categories when claimed as descriptive material *per se* (see *Warmerdam, 33 F.3d at 1360 USPQ2d at 1759)*, falling under the "process" category (i.e. inventions at that consist of a series of steps or acts to be performed). See 35 U.S.C. 100(b) ("The term process means, art, or method, and includes a new of a known process, machine, manufacture, composition of matter or material"). Functional descriptive material: "data structures" representing descriptive material *per se* or computer program representing computer listing *per se* (i.e. software per se) when embodied in a computer-readable media are still not statutory because they are not capable of causing functional change in the computer. However, a claimed computer-readable *storage* medium encoded with a data structure, computer listing or computer program, having defined structural and functional interrelationships between the data structure, computer listing or computer program and the computer software and hardware component, which permit the data structure's, listing or program's functionality to be realized, is statutory (see MPEP §2106).

Claim Rejections - 35 USC § 112

6. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

7. Claim 13 is rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

In this case, the claim discloses "realizing in each terminal of a real-time web sharing system which enables sharing of a web page via a server". It is unclear what is being realized. For the purposes of examination, the claimed clause will be interpreted as reciting, realizing in each terminal ~~of~~ a real-time web sharing system which enables sharing of a web page via a server.

Claim Rejections - 35 USC § 102

8. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

9. Claim 13 is rejected under 35 U.S.C. 102(e) as being anticipated by Kim et al. (US 2003/0105819) (referred to as Kim hereafter).

Regarding claim 13, Kim teaches a program (e.g. a component program of each remote user), comprising realizing in each terminal a real-time web sharing system which enables sharing of a web page via a server (e.g. IRC server) [0039], comprising:

a function (component program) which detects an update (e.g. scroll event) to a web page being displayed on the own terminal (e.g. an event processor receives a signal corresponding to the event [0056], wherein the event is when the client A scrolls a Web page [0078]);

a function (component program) which generates update information (e.g. control message, [0057]) which notifies the result of the update to a remote server (e.g. IRC server, [0058]) after detecting an update to a web page being displayed on the own terminal [0056]; and

a function (component program) which updates (e.g. scrolls) the web page being displayed on the own terminal [0062-[0063] based on the update information sent from said server [0060]-[0061].

Claim Rejections - 35 USC § 103

10. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

11. The factual inquiries set forth in *Graham* **v.** *John Deere Co.*, 383 U.S. 1, 148 USPQ 459 (1966), that are applied for establishing a background for determining obviousness under 35 U.S.C. 103(a) are summarized as follows:

1. Determining the scope and contents of the prior art.
2. Ascertaining the differences between the prior art and the claims at issue.
3. Resolving the level of ordinary skill in the pertinent art.
4. Considering objective evidence present in the application indicating obviousness or nonobviousness.

12. Claims 14 and 15 are rejected under 35 U.S.C. 103(a) as being unpatentable over Kim in view of Kobayashi et al. (EP 1 022 664) (referred to as Kobayashi hereafter).

Regarding claim 14, Kim teaches a program comprises

a function (component program) which updates (e.g. scroll event) a part provided on a web page on the own terminal ([0062]-[0063]) based on the part information send from the server ([0060]-[0061]).

However, Kim does not teach a function which incorporates into a web page a detecting function which detects an update to a part provided on a web page, or a function which, when said detecting function detects an update to a part provided on the web page, generates and transmits to a server the part update information which notifies the content of this update.

Kobayashi teaches a function which incorporates (modules are embedded by an embedder module, column 4 line 51-56) into said web page a detecting function (a page manager) which detects an update (detects changes) to a part provided on a web page

(a page manager monitors a state of each page element in a page, detects changes and exchanges information, column 3 lines 49-51);

a function (page manager) which, when said detecting function (page manager) detects an update (change) to a part provided on said web page, generates and transmits (exchanges information) to said server (remote page manager) part update information which notifies the content of this update (exchanges information with a corresponding remote page manager so as to dynamically perform setting of each page element to be in the same state, column 3 lines 49-54).

It would have been obvious to one of ordinary skill at the time of the claimed invention given the desirability of Kim for reliable web collaboration between two clients through use of an IRC server, the teachings of Kobayashi for sharing between browsers. One would be motivated to combine these teaching because in doing so, it would give a user the option of implementing a web sharing program within the web page, in addition to on each participating client, which would allow web sharing capabilities on devices which do not utilize the web sharing functions on a regular basis and have not installed the component program.

Regarding claim 15, a function which incorporates tags (<DIV>) for displaying a pointer to be shared on the web page between terminals (remote pointer) (Kobayashi: a remote pointer may be added to an arbitrary page, by adding it as a DIV element, column 7 lines 23-26);

a function (page controller) which, when said pointer on the web page moves, obtains and notifies to said server (Kobayashi: detects changes in a page element, communicates them to another machine, see column 2 lines 29-31);

a user machine and a server are only different in name, and it is no problem if each user machine and server consist of exactly the same hardware, (column 2 lines 43-46) the location information for the pointer after movement (Kobayashi: the pointer is moved by moving DIV to a position acquired from a mouseMove event, column 9 lines 22-23); and

a function which moves said pointer on a web page based on the location information notified from said server (when a message changing a position of a pointer is received, the position of the pointer is changed as instructed, column 9 lines 39-41).

13. Claims 1, 7, 16 and 17 are rejected under 35 U.S.C. 103(a) as being unpatentable over Kim in view of Cano et al. (US 7,127,720) (referred to as Cano hereafter).

Regarding claim 1, Kim teaches a system which enables real-time sharing of a web page being viewed on a plurality of terminals, comprising:

a means (IRC client module, [0058]) to transmit update information (control message, [0057]) which notifies an update to a web page (e.g. scroll event), such information being sent from a prescribed terminal (e.g. client A) to other terminal (client B) which is displaying the same web page as said prescribed terminal (the IRC server

receives the Web document request control message sent from the client A and transfers the received control message to the client B participating in the session opened by the client A, [0072]); and

terminals each provided with a means (e.g. message creator) to generate update information [0057] which notifies the result of the update [0056] and to transmit the resultant update information to said server [0058]; and

a means (message processor, [0062]) to cause the updating script to update the web page based on said received update information (causing the Web page scroll to occur in the client B synchronously with that in the client A, [0079]).

However, Kim does not teach where the detecting and updating scripts transmitted by the server.

Cano teaches a server provided with a means to transmit (the client obtains the activation module from the server, column 11 lines 23-25) a detecting script (activation module) which detects an update to the web page (the activation module preferably provides the object IDs of the live objects on the web page to the OSS. In response, the OSS sends the activation module the stored update messages for those live objects, column 11 lines 37-40) and an updating script (activation module) which updates the web page (the activation module updates the properties of the live objects at the client as specified by the update messages, column 11 lines 40-42); and

terminals each provided with a receiving means (the server and client are in communication via conventional communications link such as those comprising the Internet, column 4 lines 41-45) to receive the detecting script and the updating script

(activation module) which are sent from said server (the client obtains the activation module from the server, column 11 lines 23-25);

a means to cause the received detecting script (activation module) to detect an update to the web page being displayed (the activation module preferably provides the object IDs of the live objects on the web page to the OSS. In response, the OSS sends the activation module the stored update messages for those live objects, column 11 lines 37-40).

It would have been obvious to one of ordinary skill in the art at the time of the claimed invention given the desirability of Kim for reliable web collaboration between two clients through use of an IRC server, the teaching of Cano for storing and routing of objects updated on a web page to a client browser. One skilled in the art of web sharing would recognize that utilizing the teachings of Cano would enhance the collaboration system/method disclosed by Kim by storing information, accessible to the server, regarding which clients require which web page updates. The server could utilize the data to manage received updates with which clients they should be transmitted to. One would be motivated to combine these teaching because in doing so web sharing collaboration capabilities would be improved by a plurality of clients synchronizing various web pages simultaneously using a single server.

Regarding claim 7, Kim-Cano teaches a terminal of a real-time web sharing system which enables real-time sharing of a web page via a remote server, comprising:

a means to receive a detecting script (activation module) which detects an update to the web page, an updating script (activation module) which updates the web page, and update information (Cano: update messages, column 11 line 39) which notifies the update to the web page, all of these scripts being sent from a server (Cano: the client obtains the activation module from the server, column 11 lines 23-25);

a means to cause said received detecting script to detect an update to the web page being displayed (Cano: the activation module preferably provides the object IDs of the live objects on the web page to the OSS. In response, the OSS sends the activation module the stored update messages for those live objects, column 11 lines 37-40) and, if any, generate update information (control message) which notifies the result of the update (Kim: the message creator creates a control message corresponding to the even contents, [0057]) and to transmit the resultant update information to said server (Kim: the IRC client module sends the control message created by the message creator to an IRC server, [0058]); and

a means to cause the updating script to update the web page based on the update information (control message) which has been received via said receiving means (Kim: upon receiving the event occurrence control message sent from the IRC server, the message processor applies a command associated with the analyzed event contents to the Web browser, [0060]-[0062]).

Regarding claim 16, Kim-Cano teaches a method of sharing a web page in real-time between a plurality of terminals by using existing browsers, comprising:

enabling sharing of a web page in real-time without requiring alteration of the web page in advance or without requiring alteration of the display or the content rewriting module of a browser (Kim: a standard IRC server is used, resulting in no additional work being required for implementation of a collaborative browsing server, [0019]), by linking a web page for sharing (client B) and a web page for control (client A) with each other by displaying these web pages on the screen divided into frames or in browsers in parent-child relationship (Kim: in response to the control message transferred from the IRC server, the client B requests the same Web document as that requested by the client A from the Web server, [0073]);

causing a script in the control frame to detect a change in the address of the shared page (Kim: when an event occurs in a Web browser, an event processor receives a signal corresponding to the event, [0056]);

assigning, at this timing, from the control frame (input source) to the sharing frame (clients) a hook function which hooks a sharing event (live objects) and identification information (object IDs) which designates what will be shared (Cano: a web server associated with an input source provides web pages to clients. The web pages contain live objects identified by object IDs, column 3 lines 26-29); if said sharing event occurs, generating and notifying from a server to the browsers on other terminals notification information which notifies the event that has occurred (Cano: an input source provides update messages to the routing network. The routing network determines the clients that have registered for the object ID in the update messages, and the routes the messages to those clients, column 3 lines 43-48); and

invoking a function which executes the function hooked to the targets having the same identification information (Cano: the activation modules at the clients receive the messages and update the properties of the live objects as specified by the data in the message, column 3 lines 47-50).

Regarding claim 17, Kim-Cano teaches a method of sharing in real-time a web page being displayed on a first terminal and a second terminal via a remote server, comprising the steps of:

said server transmitting (Cano: the client obtains the activation module from the server, column 11 lines 23-25) to the first terminal (client) a detecting script (activation module) which detects an update to a web page (Cano: the activation module preferably provides the object IDs of the live objects on the web page to the OSS. In response, the OSS sends the activation module the stored update messages for those live objects, column 11 lines 37-40) and transmitting to said second terminal (client B) an updating script (control message) which updates a web page (Kim: the IRC server transfers the received control message to the client B participating in the session, [0072]);

said first terminal (client) receiving the detecting script (activation module) sent from said server (Cano: the client obtains the activation module from the server, column 11 lines 23-25) and causing this received detecting script to detect an update to a web page (Cano: the activation module preferably provides the object IDs of the live objects

on the web page to the OSS. In response, the OSS sends the activation module the stored update messages for those live objects, column 11 lines 37-40);

said second terminal receiving the updating script (control message) sent from said server (Kim: receiving the event occurrence control message sent from the IRC server, [0060]);

said first terminal, if said detecting script detects an update to a web page (Kim: when an event occurs in a Web browser, an event processor receives a signal corresponding to the event, [0056]), generating (Kim: the message creator creates a control message, [0057]) and transmitting to said server update information which notifies the result of the update (Kim: the IRC client module sends the control message created by the message creator to an IRC server, [0058]);

said server transmitting the update information (control message) sent from the first terminal (client A) to the second terminal (client B) (Kim: the IRC server receives the Web document request control message from the client A and then transfers the received control message to the client B participating in the session opened by the client A, [0072]); and

said second terminal (client B) causing said received updating script to update the web page based on the update information (control message) sent from said server (Kim: in response to the control message transferred from the IRC server, the client B requests the same Web document as that requested by the client A from the Web server, [0073]).

14. Claim 2-6, 8-12 and 21 are rejected under 35 U.S.C. 103(a) as being unpatentable over Kim in view of Cano and in further view of Kobayashi.

Regarding claim 2, Kim-Cano teaches said server comprises

a means to transmit (Cano: the client obtains the activation module from the server, column 11 lines 23-25) update detecting script (activation module) which detects an update to a part provided on said web page (Cano: the activation module preferably provides the object IDs of the live objects on the web page to the OSS. In response, the OSS sends the activation module the stored update messages for those live objects, column 11 lines 37-40), and a part updating script (activation module) which updates a part provided on the web page (Cano: the activation module updates the properties of the live objects at the client as specified by the update messages, column 11 lines 40-43); and

a means to transmit the part update information (control message) which notifies the update to the part provided on the web page (e.g. scroll event, mouse event, keyboard event, and so forth, Kim: [0079]), such information being sent from a prescribed terminal (client A), to other terminals (client B) which are displaying the same web page as said prescribed terminal (Kim: upon receiving the scroll event control message sent from client A, the IRC server transfers the received control message to the client B participating in the same session, [0079]); and

said terminals each comprises

a means to receive the detecting script and the updating script (activation module) which are transmitted from said server (Cano: the client obtains the activation module from the server, column 11 lines 23-25); however, Kim-Cano does not teach incorporating the update detecting script into the web page.

Kobayashi teaches said server comprises

an incorporating script which incorporates this update detecting script (a page manager monitors a state of each page element in a page, detects changes, column 3 lines 49-51) into the web page (the page manager comprises a page controller and a page communicator; these two modules are embedded by an embedder module of the server, column 4 lines 51-56); and

said terminals each comprises

a means to cause the incorporating script to incorporate said update detecting script into the web page (each client computer having a page manager controlling said pages, column 10 lines 42-43), cause said update detecting script to detect an update to a part on said web page (means for detecting changes in one of its own pages, column 10 lines 48-49), and to transmit to said server part update information which notifies the content of the update (sending said changes to said node manager for sending said changes to said server, column 10 lines 49-51); and

a means to cause said part updating script to update the part provided on the web page based on said received part update information (means for receiving changes in a page of another client computer from said node manager, and reflecting said changes in one of its own pages, column 10 lines 53-56).

It would have been obvious to one of ordinary skill in the art at the time of the claimed invention given the desirability of Kim-Cano for reliable web collaboration between two clients utilizing a storage maintaining information regarding clients and their respective web pages, the teachings of Kobayashi for sharing between browsers. One would be motivated to combine these teaching because in doing so, it would not be necessary to install a collaborative browsing client program on each participating client browser in order to implement web sharing, while further embedding a page manager in the each page to direct the sharing of changes within them.

Regarding claim 3, an update to a part provided on said web page (changes in a page element)

is a scroll (changes in a scroll position of a page) or resize of said web page or an update of a value in the entry form on said web page (changes in value of text or buttons which are elements of a form) (Kobayashi: column 2 lines 33-36).

Regarding claim 4, said server (Cano: the functionality of the OSS is integrated into the routing network, web server, information provider, and/or dynamic content provider, column 10 lines 33-36) comprises

a storing means to store identification information (object IDs), which identifies said terminals individually (Cano: the routing network maintains a registry indicating which clients have registered for which live objects, column 3 lines 42-44), in

association with the update information and part update information (live objects) sent from the individual terminals (clients) corresponding to the identification information (Cano: the routing network determines the clients that have registered for the object ID in the update messages, column 3 lines 45-47);

a means to cause said storing means to store said update information and part update information in association with the identification information of said terminals (Cano: the routing network maintains the mappings between the live objects and the clients that are currently displaying them, column 3 lines 21-24);

a means to, when a prescribed terminal logs in using said identification information (Cano: the activation module establishes a connection with an object state storage and sends it at least some of the object IDs on the web page, column 3 lines 31-35), retrieve from said storing means the update information and part update information associated with the same identification information as said login identification information (Cano: the OSS looks up the received object IDs in its storage of current update messages for the live objects, column 3 lines 35-37); and

a means to first transmit said retrieved update information and then transmit said retrieved part update information to said prescribed terminal (Cano: the OSS sends the stored update messages for the identified objects to the client, column 3 lines 36-38).

Regarding claim 5, said server comprises

a means to, when receiving a connection request which requests a connection (session) from a prescribed terminal (client A) to other terminal (client B), transmit said

connection request to such other terminal (client B) (Kim: client A opens the session to the IRC server, the client B participates in the session opened by the client A, [0068]);

a means to, when receiving from said other terminal a notification that said other terminal is ready to respond to the connection request (Kobayashi: pressed a start of sharing button provided on a page, a process of sharing is started, column 7 lines 40-42), retrieve from said storing means the update information and part update information associated with the identification information of said prescribed terminal (Cano: the OSS looks up the received object IDs in its storage of current update messages for the live objects, column 3 lines 35-37; and

a means to first transmit said retrieved update information and then transmit said retrieved part update information to said prescribed terminal (Cano: the OSS sends the stored update messages for the identified objects to the client, column 3 lines 36-38).

Regarding claim 6, said server (Kobayashi: a user machine and server are only different in name, and it is no problem if each user machine and server consist of exactly the same hardware, column 2 lines 43-46) comprises

a means to transmit a pointer script which incorporates tags (<DIV>) for displaying a pointer to be shared on the web page between the terminals (Kobayashi: the position of the new pointer is sent to the other node, column 9 lines 11-12) and which obtains the movement location for the pointer (Kobayashi: the pointer is moved by moving DIV to a position acquired from a mouseMove event, column 9 lines 22-23), and a moving script which moves the pointer (Kobayashi: column 9 lines 15-20); and

a means to transmit the location information which notifies the movement location for the pointer on the web page, such information being sent from the prescribed terminal, to other terminals which are displaying the same web page as said prescribed terminal (a page controller consisting of scripts for controlling each page element and a page communicator consisting of applets for communicating with node managers through a message queue, column 7 lines 55-58); and

said terminals (user machine) (Kobayashi: a user machine and server are only different in name, and it is no problem if each user machine and server consist of exactly the same hardware, column 2 lines 43-46) each comprises

a means to receive the pointer script and location information which are sent from said server (Kobayashi: receives changes in a page of another computer and then reflects the same changes to its own page element, column 2 lines 29-33);

a means to cause said pointer script to incorporate the tags for sharing the pointer into the web page (Kobayashi: a remote pointer may be added to an arbitrary page by adding it as a DIV element, column 7 lines 23-28), obtain the location of the pointer after movement (Kobayashi: when a message changing a position of a pointer is received, the position of the pointer is changed as instructed, column 9 lines 39-41), and transmit to said server the location information which notifies the location thus obtained (Kobayashi: the position of the new pointer is sent to the other node, column 9 lines 11-12); and

a means to cause said moving script to move the pointer on the web page based on said received location information (Kobayashi: the pointer is moved by moving DIV to a position acquired from a mouseMove event, column 9 lines 22-23).

Regarding claim 8, said terminal (user machine) (Kobayashi: a user machine and server are only different in name, and it is no problem if each user machine and server consist of exactly the same hardware, column 2 lines 43-46) comprises

a means to receive (Kobayashi: communication manager, column 3 lines 42-43) an update detecting script (page manager) which detects an update to a part provided on said web page (Kobayashi: a page manager monitors a state of each page element in a page, detects changes, and exchanges information, column 3 lines 49-51), an incorporating script which incorporates this update detecting script into the web page (Kobayashi: these two modules are embedded by an embedder module, column 4 lines 51-56), and a part updating script (message) which updates a part provided on the web page, all of these scripts being sent from a server (Kobayashi: e.g. when a message changing value of a form element is received, the form element is changed as instructed, column 9 lines 35-37);

a means to cause the incorporating script to incorporate (embed) said update detecting script (page manager) into the web page (Kobayashi: these two modules are embedded by an embedder module, column 4 lines 51-56), cause said update detecting script (page manager) to detect an update (change) to a part on said web page, and to generate and transmit (send) to said server part update information which notifies the

content of the update (Kobayashi: page manager comprises: means for detecting changes in one of its own pages and sending said changes to node manager for sending said changes to server, column 10 lines 46-51); and

a means to cause said part updating script to update the part provided on the web page based on said received part update information (Kobayashi: means for receiving said changes in a page of another client computer from said node manager and reflecting said changes in one of its own pages, column 10 lines 53-56).

Regarding claim 9, this claim comprises substantially the same limitations and those discussed on claim 3, same rationale of rejection is applicable.

Regarding claim 10, said terminal (node) comprises

a means to receive a pointer script, to be sent from said server (Kobayashi: the communication manager and the node manager which are components of a collaboration server are connected by a Java socket. This socket is used to exchange information for sharing among nodes, column 7 lines 49-53), which incorporates tags (<DIV>) for displaying a pointer to be shared on the web page between the terminals (Kobayashi: remote pointer may be added to an arbitrary page, by adding it as a DIV element, column 7 lines 23-28), and which obtains the movement location for the pointer (Kobayashi: the position of the new pointer is sent to the other node, column 9 lines 11-12), a moving script which moves the pointer (Kobayashi: the pointer is moved by moving DIV to a position acquired from a mouseMove event, column 9 lines 22-23),

and location information which notifies the movement location for the pointer on the web page (Kobayashi: a message changing a position of a pointer is received, column 9 lines 39-40);

a means to cause said pointer script to incorporate the tags (<DIV>) for sharing the pointer into the web page (Kobayashi: remote pointer may be added to an arbitrary page, by adding it as a DIV element, column 7 lines 23-28), obtain the location information for the pointer after movement (Kobayashi: a message changing a position of a pointer is received, column 9 lines 39-40), and transmit to said server the location information which notifies the location thus obtained (Kobayashi: sending changes to node manager for sending changes to server, column 10 lines 49-51); and

a means to cause said moving script to move the pointer on the web page based on said received location information (Kobayashi: when a message changing a position of a pointer is received, the position of the pointer is changed as instructed, column 9 lines 39-41).

Regarding claim 11, a server of a system which enables sharing of a web page being viewed between a plurality of terminals in real-time, comprising:

a means to receive the update information (control message) which notifies an update to a web page (Kim: the IRC server receives the Web document request control message sent from the client A, [0072]) and the part update information (control message) which notifies an update (e.g. scroll event) to a part provided on the web

page (Kim: receiving the scroll event control message sent from client A, [0079]), both the update information being sent from a prescribed terminal (client A);

a storing means to store identification information which identifies said prescribed terminal (client), in association with said received update information and part update information (live objects) (Cano: the routing network maintains the mappings between the live objects and the clients that are currently displaying them, column 3 lines 21-23);

a means to cause said storing means to store said identification information which identifies said prescribed terminal (client), in association with said update information and part update information (Cano: the clients contact the routing network and register for update messages for the object IDs. The routing network, in turn, preferably maintains a registry indicating which clients have registered for which object IDs, column 5 lines 17-21);

a means to, when other terminal logs in using the identification information of said prescribed terminal (Cano: the activation module establishes a connection with an object state storage and sends it at least some of the object IDs on the web page, column 3 lines 31-35), retrieve from said storing means the update information and part update information (updates to the properties of live objects) associated with the same identification information (object IDs) as said login identification information (Cano: the functionality of the OSS is integrated into the routing network and/or server, column 10 lines 33-36) (Cano: the OSS looks up the received object IDs in its storage, column 3 lines 35-42); and

a means to first transmit said retrieved update information and then transmit said retrieved part update information to said other terminal (Cano: the OSS sends the stored update messages for the identified objects to the client, column 3 lines 36-38).

Regarding claim 12, said server comprises

a means to, when receiving a connection request which requests a connection (session) from a prescribed terminal (client A) to other terminal (client B), transmit said connection request to such other terminal (Kim: if the client A opens the session to the IRC server, then the client B participates in the session opened by the client A, [0068]);

a means to, when receiving from said other terminal a notification that said other terminal is ready to respond to the connection request (Kobayashi: when a user has logged in on a page which indicates the start of sharing or has pressed a start of sharing button provided on a page, a process of sharing is started, column 7 lines 40-42), retrieve from said storing means the update information and part update information (updates to the properties of live objects) associated with the identification information (object IDs) of said prescribed terminal (Cano: the functionality of the OSS is integrated into the routing network and/or server, column 10 lines 33-36) (Cano: the OSS looks up the received object IDs in its storage, column 3 lines 35-42); and

a means to first transmit said retrieved update information and then transmit said retrieved part update information to said prescribed terminal (Cano: the OSS sends the stored update messages for the identified objects to the client, column 3 lines 36-38).

Regarding claim 21, comprising the steps of:

said server transmitting a pointer script, which incorporates tags (<DIV>) for displaying a pointer to be shared (remote pointer) on the web page between the terminals (Kobayashi: a remote pointer may be added to an arbitrary page, by adding it as a DIV element, column 7 lines 23-28) and which obtains the movement location for the pointer, and a moving script which moves the pointer (Kobayashi: a pointer is moved by moving DIV to a corresponding position with a mouseMove event to a window, column 7 lines 36-37);

said first terminal receiving the pointer script sent from said server (Kobayashi: a pointer process is performed, column 9 lines 9-11);

said second terminal receiving the moving script sent from said server (Kobayashi: the position of the new pointer is sent to the other node, column 9 lines 11-12);

said first terminal causing said received pointer script to incorporate the tags (DIV) of the pointer to be shared between the terminals into said web page (Kobayashi: a pointer can be added to any HTML page as a DIV element, column 9 lines 12-20) ;

said first terminal causing said pointer script to obtain the movement location for said pointer (Kobayashi: the pointer is moved by moving DIV to a position acquired from a mouseMove event, column 9 lines 22-23) and transmit the location information thus obtained to said server (Kim: a control message corresponding to the scroll event is sent to the IRC server, [0078]);

said server forwarding the location information sent from said first terminal to said second terminal (Kim: upon receiving the scroll event control message sent from the client A, the IRC server transfers the received control message to the client B participating in the same session, [0079]); and

said second terminal causing said moving script to move the pointer being displayed on said web page, based on the location information sent from said server (Kobayashi: when a message changing a position of a pointer is received, the position of the pointer is changed as instructed, column 9 lines 39-41).

15. Claim 18-20 are rejected under 35 U.S.C. 103(a) as being unpatentable over Kim in view of Cano and in further view of Ohkado et al. (Pub. No.: US 2001/0016873) (referred to as Ohkado hereafter).

Regarding claim 18, comprising the steps of:

said server transmitting to the first terminal (client) an update detecting script (activation module) (Cano: the client obtains the activation module from the server, column 11 lines 23-25) which detects an update to a part provided on a web page (Cano: the OSS sends the activation module the stored update messages for those live objects, column 11 lines 38-40) and an incorporating script (embedding module) which incorporates this update detecting script (page manager) into the web page (Kobayashi: a page manager is embedded in an HTML page by an embedding module of the server, column 7 lines 53-55) and transmitting to the second terminal an updating script which

updates a web page (Kim: the IRC server receives the Web document request control message from the client A and then transfers the received control message to the client B participating in the session opened by the client A, [0072]);

said first terminal receiving the detecting script and incorporating script sent from said server (Ohkado: when the customer information terminal acquires the content information from the Web server via the collaboration server: (1) a second customer applet which detects a change in content information displayed by the customer browser, [0071]);

said second terminal receiving the updating script sent from said server (Ohkado: the collaboration server embeds therein a client controller for displaying a page corresponding to the changed page information of the other part with which a collaboration is executed, [0009]);

said first terminal causing said received incorporating script to incorporate said update detecting script into said web page (Ohkado: the Web browsers are capable of executing applets acquired from the collaboration server and the Web server, [0122]);

said first terminal, if the update detecting script detects an update to said part provided on the web page (Kim: when an event occurs in a Web browser, an event analyzer analyzes the contents of the event, [0056]), generating and transmitting to said server part update information which notifies the content of this update (Kim: the message creator creates a control message corresponding to the event contents from the event analyzer and provides the created control message to the IRC client module,

[0057]; the IRC client module sends the control message created by the message creator to the IRC server in the IRC server system over an Internet network, [0058]);

said server transmitting the part update information (control message) sent from said first terminal (client A) to the second terminal (client B) (Kim: upon receiving the scroll event control message sent from the client A, the IRC server transfers the received control message to the client B participating in the same session, [0079]);

and said second terminal causing said received updating script to update the part provided on the web page based on the part update information which has been sent from said server (Kim: causing the Web page scroll to occur in the client B synchronously with that in client A, [0079]).

It would have been obvious to one of ordinary skill in the art at the time of the claimed invention given the desirability of Kim-Cano for reliable web collaboration between two clients utilizing a storage maintaining information regarding clients and their respective web pages, the teaching of Ohkado for collaborating a customer terminal and an agent terminal through use of a collaboration server. One would be motivated to combine these teaching because in doing so, it would eliminate the installation of special software beforehand and collaboration would take place regardless of events occurring in the information terminals.

Regarding claim 19, comprising:

using an additional third terminal; and

having the steps of:

said server storing the identification information (object IDs) of said first terminal in association with said received update information and part update information (live objects) (Cano: the routing network maintains the mappings between the live objects and the clients that are currently displaying them, column 3 lines 21-23);

said third terminal logging into said server using the same identification information (object IDs) as the identification information of said first terminal (Cano: the routing network maintains a registry indicating which clients have registered for which live objects, column 3 lines 42-44; this teaches that a plurality of clients can register for the same object IDs);

said server retrieving the update information and part update information associated with the same identification information as the said login identification information (Cano: the OSS looks up the received object IDs in its storage, column 3 lines 35-37);

said server first transmitting said retrieved update information and then transmitting said retrieved part update information (Cano: sends the stored update message for the identified objects to the client, column 36-38);

and said third terminal first updating the web page and then updating the part on the web page, based on the update information and part update information, respectively, sent from said server (the activation module at the client updates the properties of the live objects as specified by the update messages, column 3 lines 37-40).

Regarding claim 20, comprising the steps of:

said first terminal (customer) detecting that a Connect button (call button) which calls said second terminal (agent) has been pressed (Ohkado: when the customer clicks the call button and requests initiation of collaboration, the request includes a URL without a host name portion, which specifies an original page, [0159]);

said first terminal (customer), when said Connect button is pressed, notifying said server a connection request which requests a connection with said second terminal (agent) (Ohkado: when the customer enters predetermined information and requests the start of the collaboration with the agent, the applet C transmits an agent assignment request including the information entered by the user and UAI to the call manager via the session management section on the collaboration server, [0164]);

said server, when receiving the notification of the connection request from said first terminal, transmitting this notification to second terminal (Ohkado: the call manager prepares the entry of the customer queue, and enters the entry to the queue of the queue server, [0165]);

said second terminal (agent), when receiving the notification of the connection request from said server, enabling a Respond button (log-on button) which responds to this connection request and detecting the pressing of said Respond button (Ohkado: when the agent requests the initiation of a log-on processing by clicking the log-on button, the request is sent to the UAI generation section, [0150]);

said second terminal, when detecting the pressing of said Respond button, notifying said server that the terminal is ready to respond to the connection request (Ohkado: when the agent clicks a button for instructing log-on in the control window, the applet A transmits the information entered by the agent and the log-on information including the UAI to the session management section on the collaboration server, [0154]);

said server, when receiving the notification from said second terminal that the terminal is ready to respond to the connection request (Ohkado: if the agent corresponding to the queue name is in a waiting state, [0165]), retrieving the update information and part update information associated with the identification information of said first terminal (Ohkado: transmitting the changed content specifying information to the collaboration server, [0039]);

said server first transmitting said retrieved update information and then transmitting said retrieved part update information (Ohkado: the collaboration server to transmit the changed content specifying information to the second information terminal, [[0040]); and

said second terminal first updating the web page and then updating the part on the web page, based on the update information and part update information, respectively, sent from said server (Ohkado: causing the second information terminal to acquire the changed content specifying information, [0041]).

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to MADHU KHANNA whose telephone number is (571)270-3629. The examiner can normally be reached on Mon-Thurs 7:30am-5:00pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Beatriz Prieto can be reached on 571-272-3902. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/M. K./
Examiner, Art Unit 4117

/Prieto, Beatriz/
Supervisory Patent Examiner, Art Unit 4117